# **MILESTONE 1** -- SFT221 SCRUM Report and Reflections

This report should be completed in the class and submitted at the end of class. Late submissions cannot be accepted without prior approval of the instructor.

**GROUP**: C

**Members Present**:

|  |  |
| --- | --- |
| 1. Manav Zadafiya | 4. Ashraf Bharot |
| 2. Fenil Soni | 5. |
| 3. Sunny Vavadiya | 6. |

**Milestone 1 Tasks**

In this phase of the project you will:

* Setup teams of about 3-5 developers (6 is too large)
* Write and sign a team contract
* Create a GIT account
* Create a Jira account
* Add your professor to the GIT and Jira accounts
* Update Jira with the work performed and planned

**Deliverables Due at End of Lab**

* Completed SCRUM report & reflections

**Deliverables Due 24 hours after lab**

* Completed team contract
* Fully initialized Git repository
* Fully setup Jira project

**Rubric**

|  |  |  |
| --- | --- | --- |
| **Individual** | Group Participation | 75% |
| Teamwork | 25% |
| **Group** | Contract | 15% |
| Git Repository | 25% |
| Jira Project | 25% |
| SCRUM Report & Reflections | 35% |
| **NOTE** | Both the individual and group marks are calculated separately. Each member of the group will have their mark calculated based on their contribution to the group work and their contributions to the team. The group participation is a percentage that your professor feels you contributed to the group work. This is multiplied by the weight of the group participation component to determine your grade. |  |

**SCRUM Report**

**Summary of Tasks Completed or Delayed in the last week:**

Here you can list all of the tasks completed in the last week along with any tasks which could not be completed with a reason why they could not be completed.

|  |  |  |
| --- | --- | --- |
| **Member** | **Tasks Completed** | **Tasks Delayed/Blocked** |
| **Manav** | **Setup Github for project, Jira, GitTortoise and group contract** |  |
| **Fenil** | **Create Github, Jira account and setup GitTortoise, review group contract** |  |
| **Sunny** | **Create Github, Jira account and setup GitTortoise, review group contract** |  |
| **Ashraf** | **Create Github, Jira account and setup GitTortoise, review group contract** |  |
|  |  |  |
|  |  |  |
|  |  |  |

For every task delayed or blocked, describe the reason for the delay or block, how it impacts the project and the proposed solution or workaround**.**

|  |  |
| --- | --- |
| **Delayed or Blocked Task** |  |
| **Reason for delay or block** |  |
| **Impact on Project** |  |
| **Solution or work-around** |  |
|  |  |
| **Delayed or Blocked Task** |  |
| **Reason for delay or block** |  |
| **Impact on Project** |  |
| **Solution or work-around** |  |

**Summary of Meeting:**

A summary of the main points discusses in the meeting and the outcomes of the discussions.

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Team Lead | **Deciding who will represent team** | **Manav will be team leader** |
| Group Contract | **Defining terms and conditions for project** | **Contract signed** |
| Github setup | **How to create and setup git and Github** | **Setup completed** |
| Jira setup | **How and for what tasks we are using Jira** | **Setup completed** |
| Git tortoise setup | **How to setup git tortoise** | **Setup completed** |
| Project | **Review submission dates and deliverables to create project plan** | **Project Planning started** |
|  |  |  |

**Summary of Decisions Made:**

This will include major architecture and design decisions, testing decisions, prioritization of tasks, dealing with problems encountered and other major outcomes from the meeting.

|  |  |
| --- | --- |
| Decision | Rationale |
| Task distribution | Assigning equal tasks to everyone in the group to meet productivity |
| Weekly meetings | Minimum of 3 meetings to meet the project requirements and may schedule additional meeting if needed. |
| Assistance | As working in team, assisting team member who stuck at something can help to achieve our goals. |
| Deadlines | Everyone has to strictly follow deadlines as it is crucial part of project development. |
|  |  |
|  |  |
|  |  |

**Tasks Attempted During Meeting:**

Each member is assumed to participate in the SCRUM meeting and contribute to the completion of the SCRUM report and reflections. Since the SCRUM meeting will not take more than 20-30 minutes, there is lots of time left to undertake some of the actual work tasks. In the table below, each member should list what they did to complete the SCRUM report, the reflections, and 1-4 other tasks they completed during the class period. If a task could not be completed, the student should indicate why this was not possible.

|  |  |  |  |
| --- | --- | --- | --- |
| Member | Task Attempted | Time Spent | Complete? |
| ALL | **Group Contract** | **20 min.** | **Yes** |
| ALL | **Github setup** | **10 min.** | **Yes** |
| ALL | **Jira setup** | **25 min.** | **Yes** |
| ALL | **Git tortoise** | **5-10min.** | **Yes** |
| ALL | **Project discussion** | **30 min.** | **Yes** |
|  |  |  |  |
|  |  |  |  |

**SCRUM Tasks Selected for Next Week**:

The tasks each member has selected to pursue for this class or the next week.

|  |  |
| --- | --- |
| Group Member | Task Description |
| ALL | Review full project and get familiar with git, Github and Jira user-interfaces |
| ALL | Next Meeting will be on Tuesday, 21st Oct, and review milestone 2 |
| Ashraf | Review all source code: main.c, mapping.h and mapping.c |
| Fenil | Review all source code: main.c , mapping.h and mapping.c |
| Sunny | Analysis problem |
| Manav | Create Data structure |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

This is where you should highlight the major accomplishments of the class.

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Setting-up Github, git, Jira, and git tortoise | **Base of the project which enables this project** |
| Group Contract | **Clear terms and conditions help team to work effectively** |
| 1st Scrum report | **Everyone gets familiar with scrum report** |
|  |  |
|  |  |
|  |  |

**Things That Went Well in This Meeting:**

Here you can highlight things which worked well. This indicates that the way you worked on these items is working and should be continued.

|  |  |
| --- | --- |
| Topic/Work Item | Reason for Success |
| Project Setup | **Contribution from all team members** |
| Meeting and work to be done | **Dedication of team members towards project** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Things That Did NOT go Well in This Meeting:**

This is where you can list things which did not go well in the class. You should analyze why this happened and suggest how you can improve it next time. This will lead to the goal of *continuous process improvement*.

|  |  |
| --- | --- |
| Topic/Work Item | Reason for Problem and How to do Better |
| Time management | **As we all are new to Git, Github and Jira. It takes a lot more time than expected to explore these tools. However, will everyone’s support and collaboration we did well. And from next time, we will come up with preparation for meeting to make it more effective.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Reflections (to be answered by the group)**:

1. GIT is an example of a version control system. List and explain 3 benefits of using a version control system.

GIT is a very popular, widely used, and open-source tool used to track projects. There are many benefits of using git that are:

Collaboration and Teamwork: version control system enables seamless collaboration among team members. Multiple people can work on the same project simultaneously without interfering with each other's work. Each collaborator can create their own branch to develop new features or make changes independently. This ensures that work progresses smoothly, reduces conflicts, and enhances productivity.

Version Tracking and History: Version control systems maintain a comprehensive history of changes made to files. This is invaluable for tracking the evolution of a project. One can easily revert to a previous version of a file or the entire project if something goes wrong. This capability is particularly useful when debugging issues or when one needs to understand how and when specific changes were introduced. It provides a safety net for your project's integrity.

Backup and Disaster Recovery: Version control systems act as a robust backup mechanism. All project files and their entire history are stored in a repository. If someone accidentally delete a file or in case of computer crashes, one can retrieve the latest version from the repository. This redundancy and backup feature ensures that one’s work is safeguarded, reducing the risk of data loss and the need for complex backup solutions.

To sum up all, version control systems simplify collaboration, help us track changes, and provide a safety net for our project's files. They are indispensable tools for both individual developers and teams, making development more efficient and secure.

1. Jira is a modern, web-based tool for managing software projects. Describe 3 advantages of using a project management tool like Jira.

Project management tools are software or systems used to plan, organize, and oversee projects. They help teams and individuals manage tasks, resources, timelines, and communication to achieve project goals efficiently. These tools are used to:

Plan and Schedule: Define project tasks, assign responsibilities, and create timelines.

Track Progress: Monitor task completion, timelines, and budgets in real-time.

Collaborate: Facilitate team communication, file sharing, and coordination.

Resource Management: Allocate and manage resources effectively.

Reporting: Generate reports and insights for decision-making.

Overall, project management tools improve organization, communication, and efficiency in project execution and help us to deliver final project seamlessly.

1. Write a brief history of the Kanban board. Describe why it is useful in a project like this one.

The Kanban board started in Japan for better manufacturing in the 1940s, and later, it moved to other fields like software development. It's a visual tool that helps teams see their work and make it more efficient.

Kanban boards are helpful in projects because it provides:

Visual Clarity: They show work progress clearly.

Real-Time Tracking: Team members can see what's happening now.

Efficiency: They reduce overloading and help people focus.

Continuous Improvement: Teams can fix problems as they arise.

Flexibility: Kanban fits different types of projects and workflows.

So, Kanban boards make work easier to manage and improve in many project situations, including software development.